

Reference 2: JP 2001-5690 A

[0025] The expected value 13 and the test result 14 saved in this way are inputted to the test result collation section 106 shown in Fig. 1. The test result collation section 106 collates image data having the same name and outputs the differential as the report 15. The name of a terminal where the differential arises, the expected value 13, the test result 14, and file names of the image data are described in the report 15. As a result, test result can be determined automatically and easily.

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-005690

(43)Date of publication of application : 12.01.2001

(51)Int.Cl.

G06F 11/28

(21)Application number : 11-174153

(71)Applicant : NEC IC MICROCOMPUT SYST LTD

(22)Date of filing : 21.06.1999

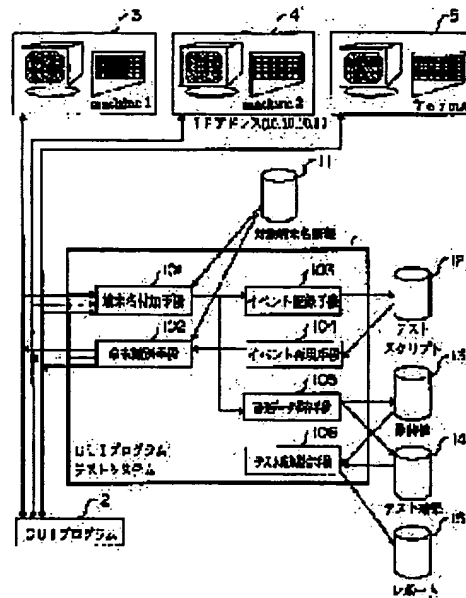
(72)Inventor : TAKAHASHI YOSHIYO

(54) PROGRAM TEST SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To make event generation and data manageable in plural terminals.

SOLUTION: In the GUI program test system 1, an execution event of an object terminal specified by objet terminal name information 11 is sampled, a terminal name adding means 101 adds a terminal name to the execution event, an image data event is stored in an image data storing means 105 as an expected value 13, event information is recorded by an event recording means, and each of events is described as functions by a test script 12. At the execution of the 2nd test and after, the test script 12 is inputted to an event reproduction means 104 to recognize the event and the terminal name, a terminal identification means 102 judges the object terminal, the screen state of the terminal is stored in the means 105 as a test result 14, the stored test result is collated by a test result collation means 106, and a difference is outputted as a report 15.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-5690

(P2001-5690A)

(43) 公開日 平成13年1月12日 (2001.1.12)

(51) Int.Cl.⁷

G 0 6 F 11/28

識別記号

3 4 0

F I

G 0 6 F 11/28

デマコード* (参考)

3 4 0 A 5 B 0 4 2

審査請求 未請求 請求項の数10 O L (全 10 頁)

(21) 出願番号

特願平11-174153

(22) 出願日

平成11年6月21日 (1999.6.21)

(71) 出願人 000232036

日本電気アイシーマイコンシステム株式
社

神奈川県川崎市中原区小杉町1丁目403番
53

(72) 発明者 高橋 佳代

神奈川県川崎市中原区小杉町1丁目403番
53 日本電気アイシーマイコンシステム株
式会社内

(74) 代理人 100089875

弁理士 野田 茂

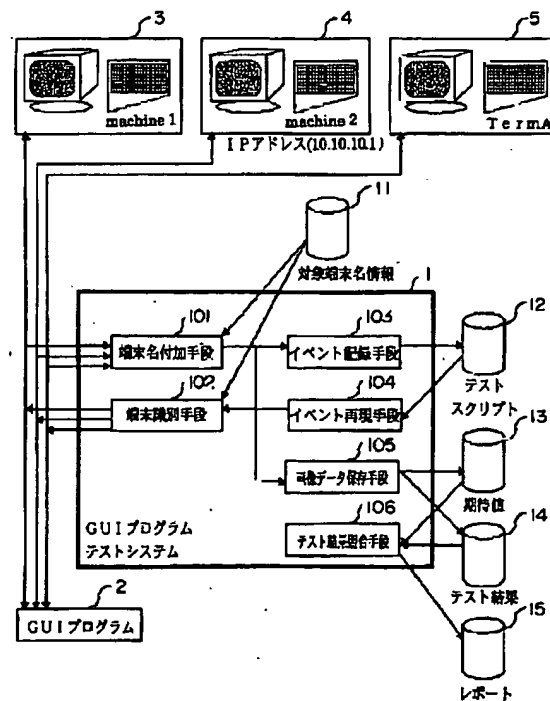
最終頁に続く

(54) 【発明の名称】 プログラムテストシステム

(57) 【要約】

【課題】 複数端末におけるイベント発生、データの管理が可能となるプログラムテストシステムを提供すること。

【解決手段】 GUIプログラムテストシステム1は対象端末名情報11で指定された対象端末の実行イベントを採取し、端末名付加手段101により端末名を付加し、画像データイベントは画像データ保存手段105で期待値13として保存し、イベントの情報はイベント記録手段103でテストスクリプト12で各イベントを関数として記述する。2回目以降のテスト実行時に、テストスクリプト12をイベント再現手段104に入力し、イベント、端末名を認識し、端末識別手段102が対象端末を判断し、端末の画面状態を画像データ保存手段105でテスト結果14として保存し、保存されたテスト結果14をテスト結果照合手段106で照合して差分をレポート15として出力する。



【特許請求の範囲】

【請求項1】 プログラムまたはシステムにより制御される端末の入力ファイルに格納された対象端末名情報を基に決定したモニタリングする端末でテスト実施者により実行されたイベントを採集してそのイベントに端末名を付加する端末名付加手段と、

上記イベントの情報を時系列にテストスクリプトに変換し、上記端末名付加手段によって付加された端末名とともに記録するイベント記録手段と、

2回目以降の上記端末のテスト実行時に上記テストスクリプトからイベント再現手段で再現した上記端末名を付加したイベントを上記対象端末名情報を参照して認識する端末識別手段と、

を備えることを特徴とするプログラムテストシステム。

【請求項2】 上記対象端末名情報は、テキスト形式で上記テストスクリプト上で使用する端末名と実際の端末ホスト名あるいはIPアドレスを1対1で記述されることを特徴とする請求項1記載のプログラムテストシステム。

【請求項3】 上記対象端末名情報は、ホスト名に対する端末に付加する端末名と、IPアドレスを有するホスト名に対する端末に付加する端末名と、上記ホスト名と異なるホスト名に対する端末に付加する端末名とを設定されることを特徴とする請求項2記載のプログラムテストシステム。

【請求項4】 上記対象端末名情報は、上記テストスクリプト記録時に使用していた対象端末名情報のファイルに記述されているホスト名のデータを変更して編集することを特徴とする請求項3記載のプログラムテストシステム。

【請求項5】 上記テストスクリプトは、C言語に似たプログラミング言語で各イベントを関数として記述することを特徴とする請求項1記載のプログラムテストシステム。

【請求項6】 上記関数は、端末名と相対移動座標を引数とするマウス移動を示す関数と、イベント内容が「端末上でマウス左ボタンをクリックする」の意味を表すマウスクリックに関する関数と、イベント内容が「端末の画面データにデータ名を付して保存する」の意味を表す画像データ保存の関数と、イベント内容が「端末上で所定キーを入力する」の意味を表すキー入力の関数と、イベント内容が「端末の画像データにデータ名を付して保存する」の意味を表す画像データ保存に関する関数とを含むことを特徴とする請求項5記載のプログラムテストシステム。

【請求項7】 上記イベントは、特定ファンクションキーの操作時に画像データ保存手段により画像データ保存イベントとして期待値として画像データが保存されるとともに、時系列に端末名が付加されてテストスクリプトに記録され、かつ2回目以降の上記端末のテスト実行時

に画像データ保存イベントと判断された場合には対象端末の画面状態がテスト結果として上記画像データ保存手段により保存されることを特徴とする請求項1記載のプログラムテストシステム。

【請求項8】 上記画像データ保存イベントは、2回目以降の上記端末のテストの実行時に画像データ保存イベントと判断された場合には対象端末の画面状態をテスト結果として上記画像データ保存手段によりテスト結果として保存され、かつ結果照合手段により上記期待値と上記テスト結果を入力して同じ名前の画像データを比較照合し、差分をレポートとして出力されることを特徴とする請求項7記載のプログラムテストシステム。

【請求項9】 上記プログラムは、複数個用いて上記端末をそれぞれ制御することを特徴とする請求項1記載のプログラムテストシステム。

【請求項10】 上記プログラムは、所定のプログラムの起動に伴い、それに関連するプログラムが起動して対象端末のテスト実行およびテストの結果を照合することを特徴とする請求項9記載のプログラムテストシステム。

【発明の詳細な説明】

【0001】

【発明が属する技術分野】この発明は、複数の端末を制御するプログラムまたはシステムに関するテストのイベントに端末名を付加し、時系列で記録・再実行することにより、従来複数端末からの指示で実施していた検査を自動的に再現することを可能とするプログラムテストシステムに関する。

【0002】

【従来の技術】GUI（グラフィカルユーザインターフェイス）プログラムのテストの自動化を実現するために、これまで以下のようないくつかのタイプのテストシステムが開発され、実用化されてきた。

(1)．テスト実行時のシーケンスを記録・再現する記録型、(2)．画面構造仕様などからテストシナリオを自動生成するシナリオ型、の二つを上げることができる。

【0003】一般的な従来技術について説明すると、GUIプログラムテストシステムは、テスト対象となるGUIプログラムと同じマシン上に搭載され、その構成は以下のブロックからなる。

(a)．テストシナリオまたはイベントを記録したテストスクリプトの作成部分、(b)．イベント再現部分、(c)．テスト結果（画像データ）の保存部分、(d)．テスト結果の照合部分、である。

【0004】特開平08-053962号公報によると、すべてのテストパターンを網羅したテストシナリオを予め自動作成することにより、テストの自動化が提案されている。一方、一般的な市販システムでは一度実行したテストのイベントをテストスクリプトとして記録

し、2回目以降のテストはテストスクリプトを読み込んでイベントを再現し、前回テスト結果との照合をすることでテストの自動化を実現している。

【0005】図8は、特開平09-223042号公報により開示された従来のソフトウェアのシステムテスト装置の構成を示すブロック図である。この図8において、システムテスト装置20が適用されるXウィンドウシステムは、一般にXサーバ28とXクライアント(図示せず)によるクライアントサーバシステムによって実現される。システムテスト装置20における構造仕様ファイル21は、ソフトウェアのGUI部に関する構造を定義するものであり、個別状態仕様ファイル22はソフトウェアのGUI部に関する各個別状態におけるGUI構成要素の属性を定義するものである。

【0006】これらの構造仕様ファイル21および個別状態仕様ファイル22は何れもテキストファイル形式になっており、利用者が作成するものである。テストシナリオ23は構造仕様ファイル21と状態遷移仕様ファイルにしたがって生成されるファイル情報であり、テストシナリオとは、テスト対象となるソフトウェアのGUI部に関する仕様を検証する際のGUI操作手順を定めたものである。

【0007】このテストシナリオ23は、構造仕様ファイル21と状態遷移仕様ファイルにしたがって、ソフトウェアのGUI部がもつすべての状態遷移とを網羅するように生成されたものであり、すべての入力イベントを少なくとも1回は必ず実行するようなシナリオである。このテストシナリオ23は利用者が編集可能なように、テキストファイル形式になっている。テストシナリオ23をテストシナリオ実行手段24で読み込み、読み込んだテストシナリオ23の内容を基に直接イベントメッセージ(マウス29やキーボード30から直接発生したのと同じ形式のイベントメッセージ)をこのテストシナリオ実行手段24で作成し、作成した直接イベントメッセージをXサーバ28に供給することにより、テスト対象のソフトウェアのGUI部をテストシナリオ23にしたがって間接的に操作する。

【0008】実行結果照合手段25は構造仕様ファイル21と個別状態仕様ファイル22を読み込み、テストシナリオ実行手段24において、実行中の被テストプログラムの各状態における各部品属性値が個別状態仕様に記載されている値と一致の有無を照合してその照合結果の内容に基づいてテキストファイル形式のテスト結果レポート26を作成する。GUI27は、このシステムテスト装置20に関する種々の動作を指示するためのユーザインタフェースである。このような従来のシステムテスト装置20においては、市販システムでは特に物理的なGUI画面の表示状態の照合テストに有効である。

【0009】

【発明が解決しようとする課題】しかし、最近では複数画

面に文字列を表示するプログラムや、ネットワークを介して画像や音声を相互通信し、デモンストレーションや会議を遠隔操作で行うなど単一端末だけでは、すべての全テストの実行が行えない複雑なシステムも開発されている。従来のテストシステムでは単一マシンもしくは端末上でのテストを前提としているため、これらのプログラムやシステムのテストが実行できないという課題が発生した。

【0010】一方、特開平07-152547号公報には、インタラクションに関する記述を集中管理してGUIソフトウェアのインタラクション変更を容易にするために、インタラクション記述手段によって記述された複数のインタラクションをプロセスごとに合成した関数であるプロセス関数コードを生成し、発生する種々のイベントのうちインタラクション記述手段で記述されたインタラクションで使用されるイベントを選択する関数であるイベント選択関数コードを生成し、インタラクション以外の部分を記述したプログラムコード、プロセス関数コード、およびイベント選択関数コードを合成することが開示されている。しかし、この公報の場合でも、上記課題を解決するに至っていない。

【0011】この発明は、上記従来の課題を解決するためになされたもので、端末名をイベントに付加することにより、複数端末におけるイベント発生とデータ管理が可能となるプログラムテストシステムを提供することを目的とする。

【0012】

【課題を解決するための手段】上記目的を達成するために、この発明のプログラムテストシステムは、プログラムまたはシステムにより制御される端末の入力ファイルに格納された対象端末名情報を基に決定したモニタリングする端末でテスト実施者により実行されたイベントを採集してそのイベントに端末名を付加する端末名付加手段と、上記イベントの情報を時系列にテストスクリプトに変換し、上記端末名付加手段によって付加された端末名とともに記録するイベント記録手段と、2回目以降の上記端末のテスト実行時に上記テストスクリプトからイベント再現手段で再現した上記端末名を付加したイベントを上記対象端末名情報を参照して認識する端末識別手段とを備えることを特徴とする。

【0013】そのため、端末名付加手段により入力ファイルに格納された対象端末名情報を基に、モニタリングする端末を決定し、その端末でテスト実施者により実行されるイベントを採取して、そのイベントに端末名を付加するとともに、イベント記録手段でイベントの情報を時系列でテストスクリプトに変換して端末名付加手段によって付加された端末名とともに記録する。2回目以降の端末のテスト実行時に、イベント再現手段によりテストスクリプトから端末名が付加されたイベントを対象端末名情報の参照のもとに認識し、対象の端末は端末識別

手段により対象端末名情報を参照して判断するようにしたので、複数端末におけるイベント発生とデータ管理が可能となる。

【0014】

【発明の実施の形態】以下、この発明によるプログラムテストシステムの実施の形態について図面に基づき説明する。図1はこの発明による第1実施の形態の構成を示すブロック図である。この図1を参照して、まず第1実施の形態の構成について説明するが、この第1実施の形態では、説明の都合上GUIを使用するプログラムテストシステムに適用した事例に基づいて説明する。

【0015】図1において、GUIプログラム2が端末3、4、5を制御して動作している。GUIプログラムテストシステム1は端末3、4、5とGUIプログラム2間のデータ情報をモニタリングしている。GUIプログラムテストシステム1は端末名付加手段101、端末識別手段102、イベント記録手段103、イベント再現手段104、画像データ保存手段105、テスト結果照合手段106から構成されている。GUIプログラムテストシステム1は、入力ファイルとして対象端末名情報11、入出力ファイルとしてテストスクリプト12、期待値13、テスト結果14、出力ファイルとしてレポート15を持つ。

【0016】次に、動作について説明する。初めにGUIプログラム2が制御する端末の一覧を対象端末名情報11にあらかじめ作成しておく。このファイルはテキスト形式で、テストスクリプト12上で使用する端末名と実際の端末のホスト名もしくはIP(Internet Protocol)アドレスを1対1で記述したものであり、そのフォーマットは、たとえば、図2に示す通りである。

【0017】この図2は図1の端末3、4、5の情報を記述したもので、1行目はホスト名が「machine1」の端末3に「term1」という端末名を、2行目はIPアドレスが「10.10.10.1」の端末4に「term2」という端末名を、3行目はホスト名が「TermA」の端末5に「term3」という端末名をそれぞれ設定していることを示している。このファイルはテスト実施者が自由に作成・編集することが可能である。

【0018】GUIプログラムテストシステムは、このファイルを元にモニタリングする端末を決定する。次に、テスト実施者がテストを実行することにより、テストの手順をテストスクリプト12として保存する。その方法を図3のフローチャートに沿って説明する。GUIプログラムテストシステム1は対象端末名情報11で指定された端末をモニタリング対象とし、それぞれの端末3、4、5でテスト実施者により実行されたキー入力、マウス移動、マウスクリックなどのイベントをステップF1にて採取する。

【0019】これらのイベントには、端末名付加手段1

01によって対象端末名情報11にしたがって端末名が付加される。たとえば、図1の端末3は図2において、端末名「term1」と設定されているため、端末3でマウス移動を行った場合、「term1」という端末名が付加される。

【0020】また、特定のファンクションキーを押下したイベントは画像データ保存イベントとして、ステップJ1で判別し、画像データ保存手段105により期待値13として画像データがファイルとして保存される。イベントの情報はイベント記録手段103によってテキストベースのテストスクリプト12に変換される。その際、端末名付加手段101によって付加された端末名とともに記録される。画面データ保存のイベントは他のイベント同様時系列にテストスクリプト12に端末名を付加して記録される。この記録操作がステップJ2でイベント終了と判断されるまで繰り返されることになる。

【0021】テストスクリプト12はC言語に似たプログラミング言語で各イベントを関数として記述する。フォーマットについて図4を用いて詳しく説明する。この図4において、1行目の「Mouse_move()」という関数はマウス移動を示す関数で、端末名、相対移動座標(x,y)を引数とする。つまり、一行目のイベント内容は「term1上でマウスをX方向100、Y方向100移動する」ということを意味する。

【0022】2行目の「Mouse_click()」という関数は、マウスクリックを示し、イベント内容は「term1上でマウス左ボタンをクリックする」となる。3行目の「Window_get()」という関数は、画像データ保存の関数で、イベント内容は「term2の画像データをdata1という名前で保存する」となる。4行目の「Key_input()」という関数は、キー入力の関数で、イベント内容は「term2上で"OK"とキー入力する」となる。5行目は3行目同様画像データ保存を示し、イベント内容は「term1の画像データをdata2という名前で保存する」となる。画像データの名前はイベント記録の際、自動発生させて名前がだぶらないように決定する。

【0023】2回目以降のテスト実行を図5のフローチャートに沿って説明する。テストスクリプト12をイベント再現手段104が入力し、イベントと端末名を認識する。その識別結果を端末識別手段102に出力する。これにより、対象となる端末は端末識別手段102が対象端末名情報11を参照して判断する。端末識別手段102がステップJ3にて、画像データ保存イベントと判断した場合は、対象となる端末の画面状態を画像データ保存手段105によりテスト結果14として保存する。

【0024】それ以外のイベントは、ステップF2にて対象となる端末上でイベントを発生させる。この操作をテストスクリプトに記述されているイベントが終了したことをステップJ4で判断するまで繰り返し行うことで

テスト実行の再現を実現する。

【0025】このようにして保存された期待値13とテスト結果14を図1におけるテスト結果照合手段106が入力して同じ名前の画像データを比較・照合し、差分をレポート15として出力する。レポート15の内容は差分が生じた端末名と期待値13、テスト結果14それぞれの画像データのファイル名が記述される。これにより、テスト結果の判断が自動的かつ簡単に行えることになる。

【0026】以上から明らかなように、この第1実施の形態では、GUIプログラム2に関するテストのイベントに端末名を付加し、時系列で記録し、再実行するようにしたので、複数端末におけるイベント発生、データ管理が可能となり、より複雑なGUIプログラムのテストの自動化が実現できることになる。

【0027】次に、この発明による他の実施の形態について説明する。上述した第1実施の形態においては、一つのGUIプログラムに対する検査の場合を説明したが、複数プログラムによる統合テストにも対応できる。図6はこの発明の第2実施の形態の構成を示すブロック図であり、この図6に示す第2実施の形態では、テスト対象のGUIシステムが3つのGUIプログラム2A、2B、2Cで構成され、それぞれのプログラムが端末3、4、5を制御している場合を例示している。

【0028】この図6において、構成の説明に際しては、図1と同一部分には同一符号を付すのみにとどめる。この図6を図1と比較しても明らかなように、この図6では、上述のようにテスト対象のGUIシステムが3つのGUIプログラム2A、2B、2Cで構成され、それぞれのプログラムが端末3、4、5を制御するようにしている点が図1に示した第1実施の形態とは異なるものである。

【0029】このように構成することにより、GUIプログラム2Aでマウスクリックを行うとGUIプログラム2Bがそれに反応して画面上にウィンドウを出す、といった動作をテストしたい場合も同様に端末3のマウスクリック・端末4の画像データを記録するだけでテストの実行・テスト結果の照合が可能となる。このような効果が得られるのは、テストイベント・結果の管理があくまで端末単位であるため、それらを制御するGUIプログラムを感知しなくてすむからである。

【0030】また、対象となる端末は対象端末名情報ファイルを編集することにより適宜変更が可能である。たとえば、テストスクリプト記録時には「TermA」という端末名を使用していたが、実際のテストは「TermB」という端末名を使用しなければならなくなった場合、図7のように対象端末名情報ファイルに記述されて

いる「TermA」の端末名を「TermB」に変更するだけで実行が可能となる。このように、テスト環境をフレキシブルに変更できるのは、対象端末名情報ファイルがテキストベースで、テスト実施者による変更が容易となっているためである。なお、上記各実施の形態では、GUIプログラムに関してのテストの場合で説明したが、この発明はシステムに関するテストにも適用できるものである。

【0031】

【発明の効果】以上のように、この発明によれば、複数の端末を制御するプログラムまたはシステムに関するテストのイベントに端末名を付加するようにしたので、複数端末におけるイベント発生、データ管理が可能となり、より複雑なGUIプログラムのテストの自動化が実現できる。

【図面の簡単な説明】

【図1】この発明によるプログラムテストシステムの第1実施の形態の構成を示すブロック図である。

【図2】この発明によるプログラムテストシステムの第1実施の形態におけるGUIプログラムが制御する端末の一覧を対象端末名情報のフォーマットを示す説明図である。

【図3】この発明によるプログラムテストシステムの第1実施の形態におけるテスト実行時のテスト手順のテストスクリプトへの保存方法を説明するためのフローチャートである。

【図4】この発明によるプログラムテストシステムの第1実施の形態におけるテストスクリプトのフォーマットの一例を示す説明図である。

【図5】この発明によるプログラムテストシステムの第1実施の形態に適用されるテスト実行再現時の動作を説明するためのフローチャートである。

【図6】この発明によるプログラムテストシステムの第2実施の形態の構成を示すブロック図である。

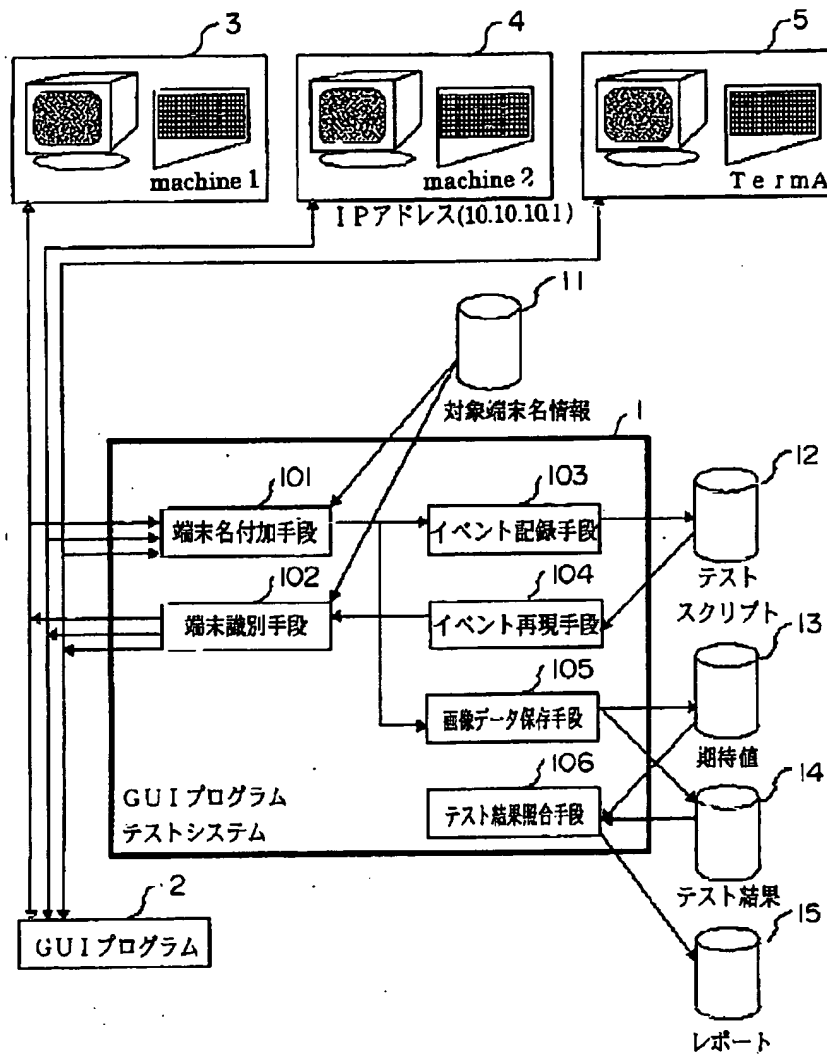
【図7】この発明によるプログラムテストシステムの第2実施の形態における対象端末変更に伴う端末名情報ファイルの変更実施の説明図である。

【図8】従来のプログラムテストシステムの構成を示すブロック図である。

【符号の説明】

1……GUIプログラムシステム、2、2A～2C……GUIプログラム、3～5……端末、11……対象端末名情報、12……テストスクリプト、13……期待値、14……テスト結果、15……レポート、101……端末名付加手段、102……端末識別手段、103……イベント記録手段、104……イベント再現手段、105……画像データ保存手段、106……テスト結果照合手段。

【図1】



【図2】

term1	machine1
term2	10.10.10.1
term3	TermA

【図4】

```

Mouse_move(term1,100,100);
Mouse_click(term1,left);
Window_get(term2,data1);
Key_input(term2,"OK"<CR>);
Window_get(term1,data2);
.....

```

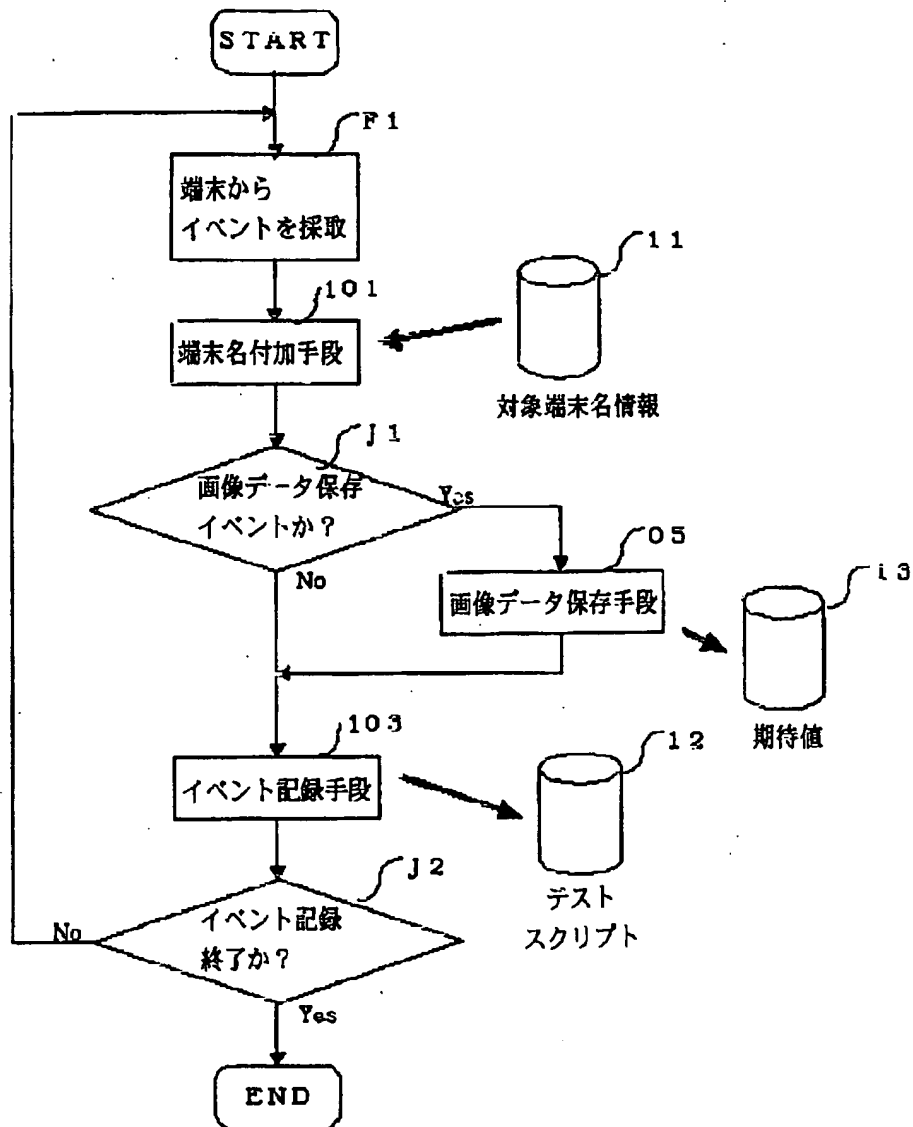
【図7】

term1	machine1
term2	10.10.10.1
term3	TermA

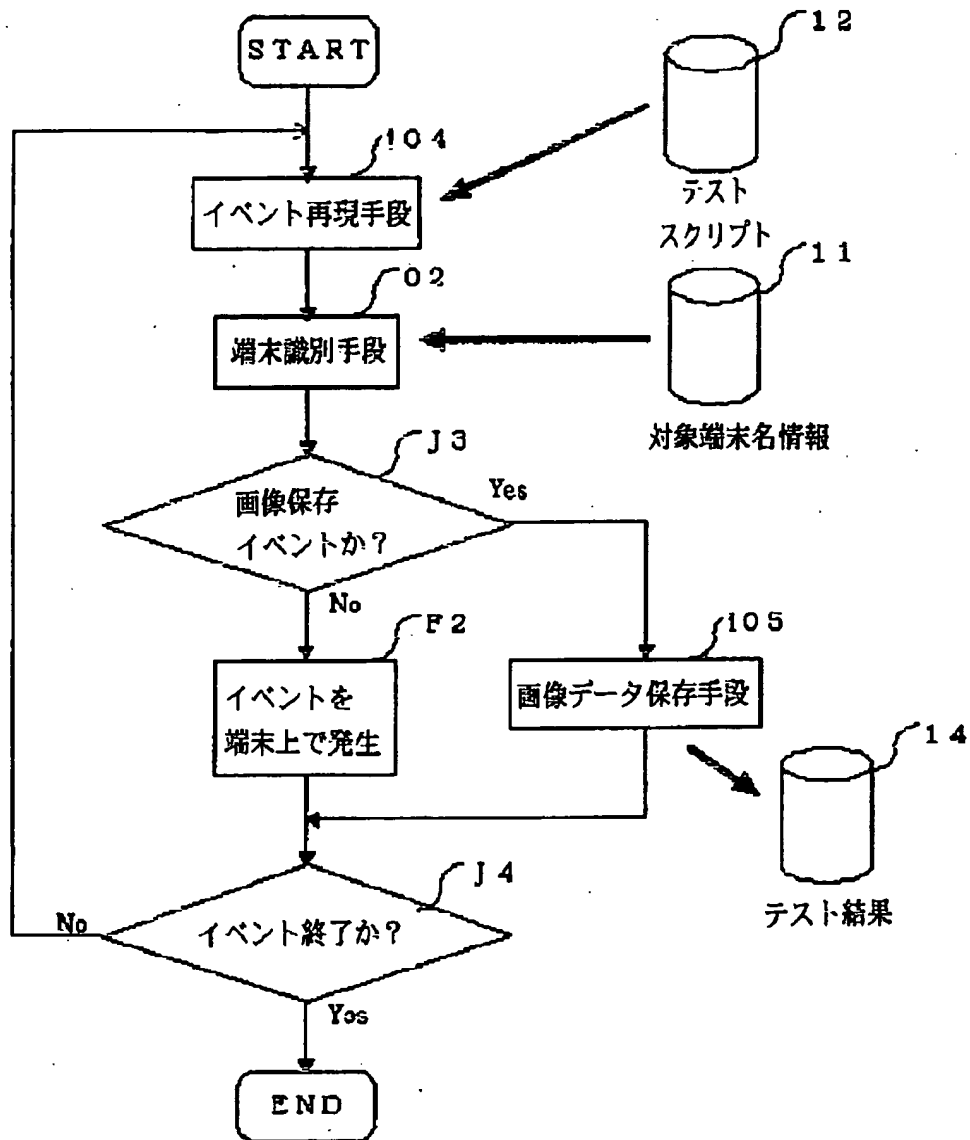
→

term1	machine1
term2	10.10.10.1
term3	TermB

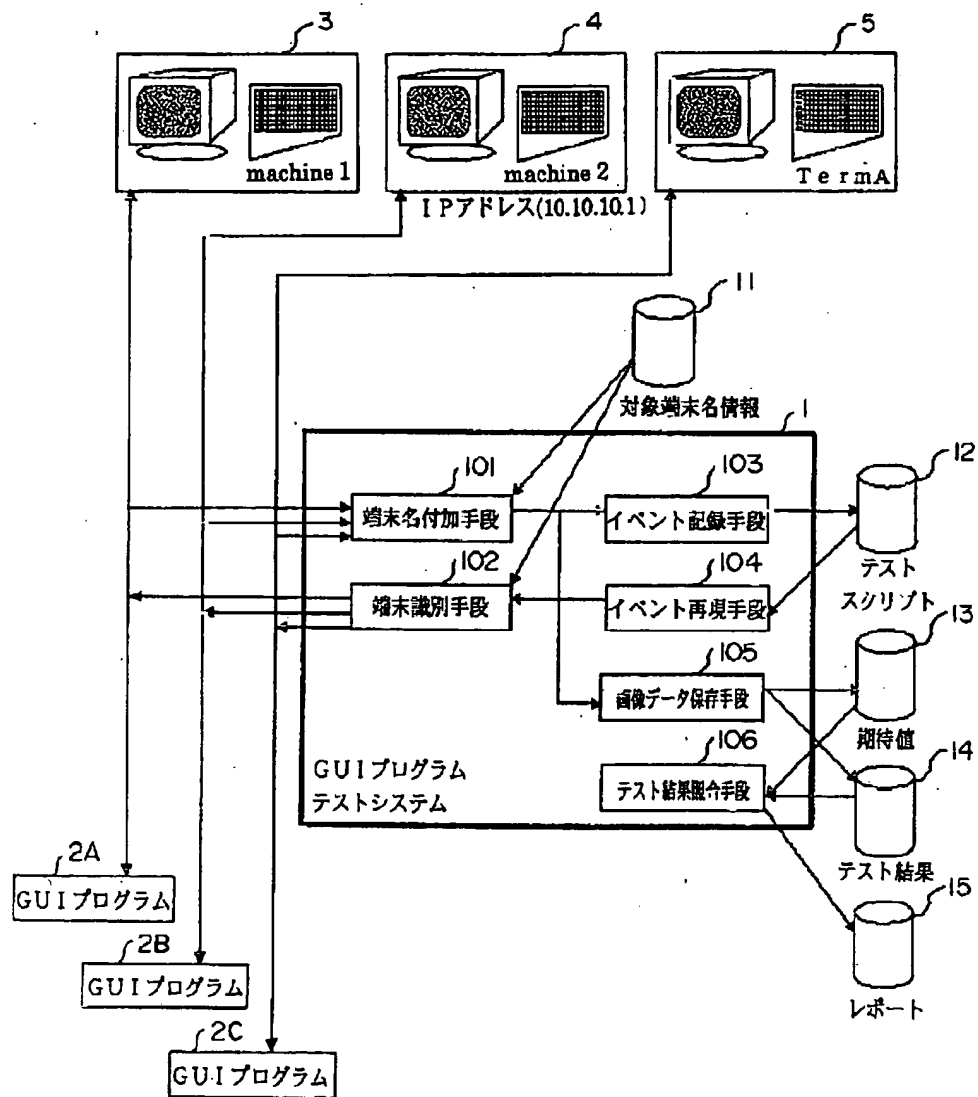
【図3】



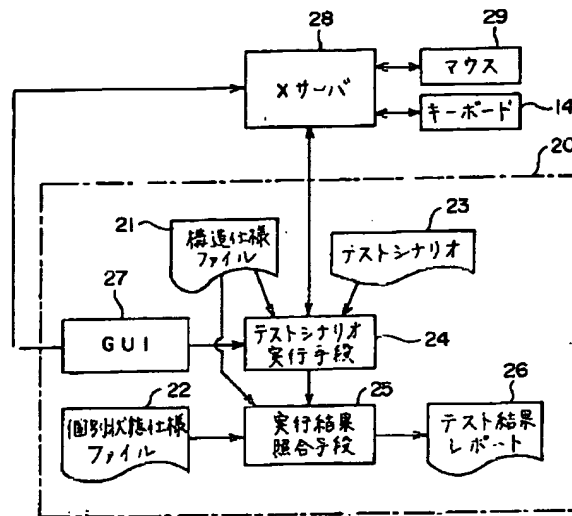
【図5】



【図6】



【図8】



フロントページの続き

Fターム(参考) 5B042 GA12 GA36 GB02 GC08 HH12
HH17 HH30 HH49 JJ30 MA08
MA14 MC08 MC35 MC37 MC40
NN04 NN23